Sensor Fault Identification on Laboratory Tower

E. ZUGASTI, P. ARRILLAGA, J. ANDUAGA, M. A. ARREGUI and F. MARTÍNEZ

ABSTRACT

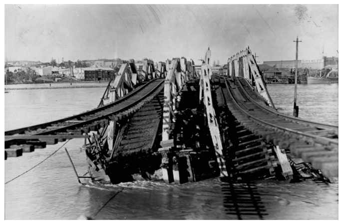
In this paper an implementation of a sensor fault identification algorithm applied to a laboratory tower is presented. The method implemented is based on information theory and has been tested in a tower demonstrator that simulates a wind turbine. Different sensor faults have been simulated, and the algorithm has successfully detected the different faults.

INTRODUCTION

Structural Health Monitoring (SHM) aims to give, at every moment during the life of a structure, a diagnosis of the "state" of the constituent materials, of the different parts, and of the full assembly of these parts constituting the structure as a whole [2]. Damages in structures have caused many disasters in the course of history as can be seen in figure 1. This has attracted the attention of the community related it to construction techniques and maintenance of structures. Among the different application fields of Structural Health Monitoring (SHM), nowadays the wind turbine one can be stuck out. The current trend in this field is to locate the wind power plants off shore, where costs, including maintenance and operations, increase significantly compared to on shore ones. This fact has increased the interest towards the implementation of different concepts of SHM in these structures.

In the figure 2a shows how the implemented method is integrated in a global SHM solution. First of all, once the data coming from the structure is stored so that damage detection method can be run. If the damage detection method tells that damage is present in the structure, it should be checked that the sensors are working correctly; that's the moment the method described in the next chapter is needed . If the sensors are OK, the alarm should be rang and the damage should be located. If there is damage in the sensor, that signal is removed, and whole process is repeated again.

Ekhi Zugasti, IK4-Ikerlan, ezugasti@ikerlan.es



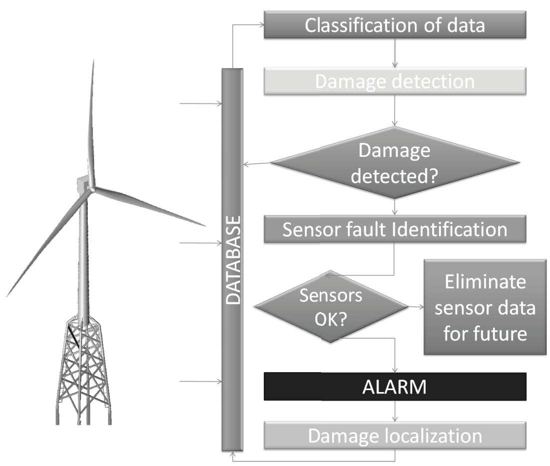
(a) *Bridge Collapse*

(b) *Airplane crash, LA*

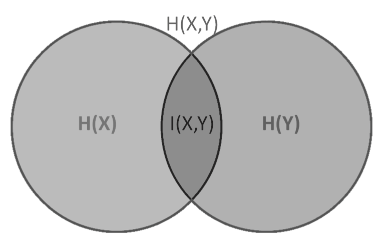
(c) *Turbine*

Figure 1: *Structural Disasters*

In order to have a secure Structural Health Monitoring (SHM) system, it is very important to know the state of our sensors. In the moment they are placed in a structure, they work correctly, but usually the lifespan of a structure is larger than the sensors lifetime. In order to know if the sensor is still healthy, in this paper we implemented a sensor fault detection algorithm that has been tested in a laboratory tower.



(a) *SHM System*



(b) *MI by entropies*

Figure 2: *SHM and Mutual Information*

This sensor fault detection method is based on Mutual Information theory. Mutual information measures the dependence between different variables; it can measure how many information is shared between two variables. This work is based on the work of Kraemer [3] and works with output time domain data, being able to identify sensor faults, searching changes in mutual information in different time stamps.

The paper is organized as follows: first, the details of algorithm are explained next, the experimental set-up and the results are showed. Finally, the conclusions are presented.

SENSOR FAULT IDENTIFICATION USING MI

Mutual Information (MI)

A good formalization of the uncertainty of a random variable is given by Shannon and Weaver's [6] information theory. Basically, the mutual information measures the amount of information contained in a variable or a group of variables, in order to predict the dependent one. It has the unique advantage to be model independent and nonlinear at the same time. Althought first it was developed for binary variables, it has been extended to continuous variables. Let X and Y be two random variables. $\mu_{X,Y}$ is the joint probability density function (pdf) of X and Y. The marginal density functions are given by

$$\mu_X(x) = \int \mu_{X,Y}(x,y)dy$$

and

$$\mu_Y(y) = \int \mu_{X,Y}(x,y)dx$$

Let us now recall some elements of information theory. The uncertainty on Y is given by its entropy defined as

$$H(Y) = -\int \mu_Y(y) log \mu_Y(y)dy$$

The same way, the uncertainty on X is given as

$$H(X) = -\int \mu_X(x) log \mu_X(x)dx$$

Using the same theory, The joint uncertainty of the (X,Y) pair is given by the joint entropy, defined as

$$H(X,Y) = -\iint \mu_{X,Y}(x,y) log \mu_{X,Y}(x,y)dxdy$$

The mutual information between X and Y can be considered as the result of the entropy of X plus the entropy of Y minus the joint entropy of X and Y

$$I(X,Y) = H(X) + H(Y) - H(X,Y)$$

This can be seen graphically in Figure 2b: the entropy of X is the first circle, the entropy of Y the second one, and the joint entropy the sum of both, that tells that the mutual information of both is the intersection of both circles.

Therefore $\mu_{X,Y}$ is only needed to estimate in order to estimate the mutual information between X and Y. And the final MI is defined as:

$$I(X,Y) = \iint \mu_{X,Y}(x,y) log \frac{\mu_{X,Y}(x,y)}{\mu_X(x)\mu_Y(y)} dxdy$$

Estimation of MI

As detailed in the previous section, estimating the mutual information (MI) between X and Y requires the estimation of the joint probability density function of (X,Y). This estimation has to be carried on the known data set. While first developed for binary variables, it has been extended to continuous variables. For this reason, the MI has to be estimated.

There are different methods to estimate the MI, Histogram-based and kernel-based pdf estimations are among the most commonly used methods [1]. However, their use is usually restricted to one-dimensional or two-dimensional probability density functions. K-nearest neighbour method is also analyzed by Kraskov [5] and is able to handle multi-dimensional probability density functions.

In this case, only two dimensional probability density functions are needed, so a method based on bins is used. This method is basically a Histogram-based estimator. The base of the logarithm determines the units in which information is measured. A forward approach is to divide the range of x and y into finite bins and to count the number of sampled pairs of $z_k = (x_k, y_k)$, $k = 1, \ldots, N$ falling into these finite bins. This count allows to approximately determine the probabilities, replacing the integrals by the finite sum.

$$I(X,Y)_{bin} = \sum_i \sum_j \mu_{X,Y}(i,j) log \frac{\mu_{X,Y}(i,j)}{\mu_X(i)\mu_Y(j)}$$

Sensor Fault Indicator

The mutual information for all possible combinations of sensor outputs y_m and y_l and (except $m = l$) is computed which leads to an MI-matrix for all combinations m, l. The basic idea is that the mutual information changes when a sensor fault f_m is present; let us say in the m-th channel:

$$\widetilde{y}_m = y_m + f_m$$

This fault appears only in the m-th channel. Thus all combinations with index m should show a reduction of the MI. This allows to localize the faulty sensor. More than one faulty sensors can be simultaneously detected in the same way.

One possibility to visualize the faulty sensor is to use the relative change in the MI as a sensor fault indicator (SFI)[4]:

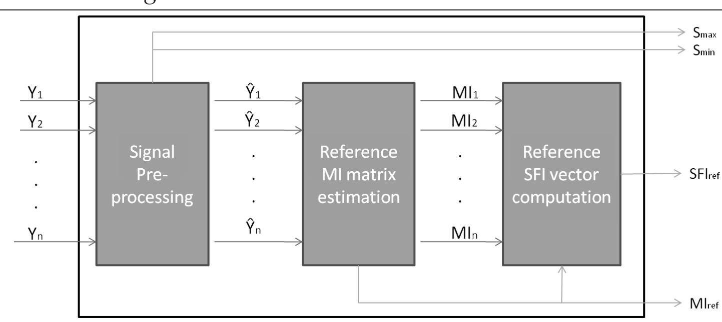
$$SFI_n = \frac{|MI_n - MI_{Ref}|}{MI_n}$$

where n is an actual data set and the lower index Ref represents one reference data set.

The Algorithm

The algorithm is divided into two phases. In the first part of the algorithm the sensors are known to be working well. In the second phase we need to identify whether the sensor works well. The algorithm provides a metric that indicates the existence of the fail in the sensors.

Algorithm 1 *Learning Phase*



Learning Phase

In the algorithm 1, the learning phase of the algorithm in a block diagram is shown. The objective of this part is to know how sensors work in a healthy condition. The different blocks are divided as: the first block, the signal will be limited, to the maximum and minimum values of each sensor, and these maximum and minimum values will be stored for the detecting phase.

Once the signal is preprocessed, the reference MI matrix is estimated (MI_{Ref}). In order to be able to do this, the Mutual Information value for each sensor pair will be estimated, and will be saved in different matrices. Once all the MI values for each pair of sensor in each different state are known, the reference Mutual Information matrix will be calculated using the mean value of all the learned cases. The reference MI matrix, will contain the mean MI values for each pair of sensors.

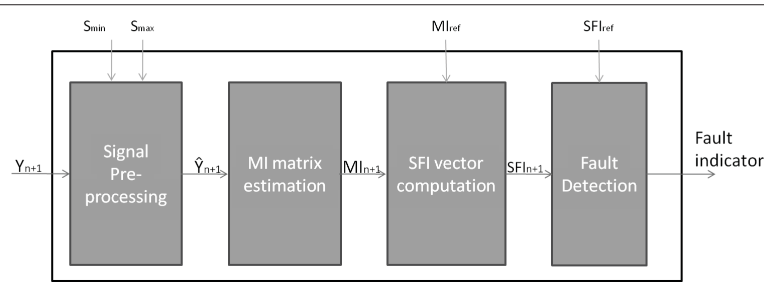
Finally, using the MI matrices coming from each data set and the reference MI, the SFI for each data set is extracted. Applying a mean value, we will have a SFI reference matrix. In order to have a good fault indicator, the sum of all the SFI values in each sensor will be made. That way, a vector is constructed, which indicates the state of each sensor. We will call that vector SFI_{Ref}.

Detection Phase

Heading to the detection phase, algorithm 2, it can be seen that most of the blocks are the same as in the learning phase. The main difference is that the input is not known to have data from healthy sensors. That is why everything used in the learning phase is needed ($S_{min}, S_{max}, MI_{ref}$ and SFI_{ref}). The first part of the detection phase is the same as the learning one. The incoming signal is limited using the limiters from the learned signal.

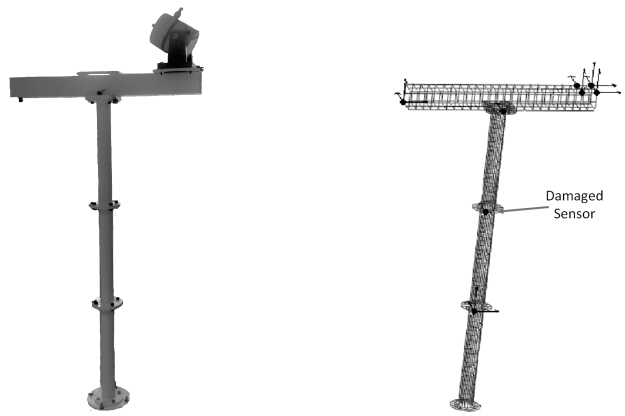
Once the signal is preprocessed, the MI matrix of each combination of sensors is estimated, and with this matrix, the Sensor Fault Indicator is calculated, transforming that matrix into a vector, as it was told before. with the reference SFI, finally the fault indicator is extracted. This fault indicator is a vector, that contains a fault indicator value for each sensor. If the difference of one or two of them is much bigger than the other values, this means that the sensor is faulty.

Algorithm 2 *Detecting Phase*



EXPERIMENTAL SETUP AND RESULTS

The tested structure is a tower model, similar to those of a wind turbine, see figure 3a. This structure is 2.2 m high; the top piece is 1 m long and 0.1 m width. This tower is composed of three sections joined with bolts. The damage is simulated by acting on these bolted joints. As it can be seen in the figure 3a there is a modal shaker placed on top of the structure, simulating the nacelle of a wind turbine, so that it can create a vibrational movement. To detect the structural response, some accelerometers are placed in the tower; in order to know the best location for these sensors, FemTools (by Dynamic Design Solutions) has been used. The pretest analysis made to the Finite Element model of the tower, resulted in 8 accelerometers, 4 triaxial, and 4 uniaxial, see figure 3b.



(a) *Photo* (b) *Sensor Locations*

Figure 3: *Tower information*

In order to simulate the sensor fault, two types of changes have been assumed. The first fault is made by changing the signal of the accelerometer directly, by adding a white noise to the properly captured signal using Matlab. The second approach has been done by acting directly in the sensor; a reusable putty-like pressure-sensitive adhesive have been attached under the sensor. That way the damping of the sensor increases, and simulates a faulty sensor.

Noise Addition Fault Result

These are the fault indicators for each sensor. As it can be seen the 4th channel has been changed, and the noise has been added to the original signal. The algorithm is able to detect the faults. The fault indicator in the 4th channel of the figure 4 is much higher than the other, so that means that the value of Mutual Information for that channel changed a lot comparing it with the rest of the channels.

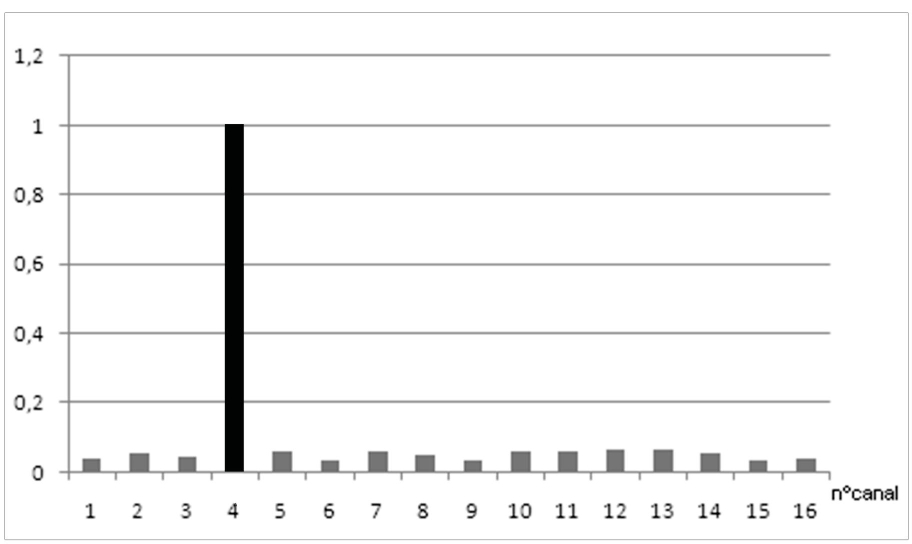


Figure 4: *Fault indicators for Noise addition fault*

Adhesive Change Fault Result

Like before, the fault indicators are for each sensor. All the sensors except the one on the 4th channel have been correctly placed, while the one on the 4th channel have been placed with the special adhesive. As the figure 5 shows, the results have been satisfactory. The fault indicator on the faulty sensor is 4 times bigger than on the other sensors. We have also attached more adhesive, and the error gets bigger, so we can say that the algorithm works well for this case too.

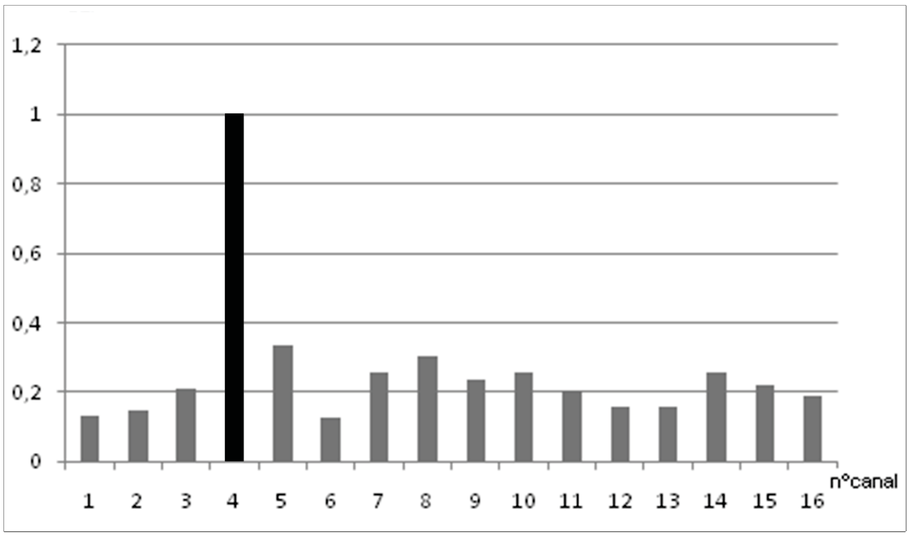


Figure 5: *Fault indicators for Adhesive change fault*

CONCLUSIONS AND FUTURE WORK

In this paper a method for sensor fault identification have been implemented, and it has been integrated into a global SHM solution. This method is based on the information theory, and more precisely, on Mutual Information. Mutual Information analyzes the dependence between sensors information. If that information dependence changes between 2 different states, we can say that the sensor is faulty. In order to have that information, the dependence between two healthy sensors must be known, that is why a learning phase is needed.

The method has been validated with the help of a tower model, and it is able to detect the faulty sensors by simulating change in the signal, and also by acting on the hardware (the accelerometer in this case).

But we know that the readings from the sensors are different when the Environmental and Operational Conditions (EOC) are different. So, if a field working solution is needed, the EOC of the structure to be monitored monitor should be taken into account.

REFERENCES

[1] R. Battiti, "Using mutual information for selecting features in supervised neural net learning.," *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*, vol. 5, pp. 537–50, Jan. 1994.

[2] C. R. Farrar and K. Worden, "An introduction to structural health monitoring.," *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, vol. 365, pp. 303–15, Feb. 2007.

[3] P. Kraemer and C. Fritzen, "Sensor Fault Identification Using Autoregressive Models and the Mutual Information Concept," *Key Engineering Materials*, vol. 347, pp. 387–392, 2007.

[4] P. Kraemer and C. Fritzen, "Sensor Fault Detection and Signal Reconstruction using Mutual Information and Kalman Filters," in *International Conference on Noise and Vibration Engineering*, pp. 3267–3282, 2008.

[5] A. Kraskov, H. Stogbauer, and P. Grassberger, "Estimating mutual information," *Physical Review E*, vol. 69, pp. 1–16, June 2004.

[6] C. E. Shannon, "The mathematical theory of communication. 1963.," *M.D. computing : computers in medical practice*, vol. 14, no. 4, pp. 306–17, 1948.